REMARKS

In this Office Action, the Examiner rejected Claims 14 – 21 under 35 U.S.C. §101 as being directed to non-statutory subject matter. Claims 1 – 31 were rejected under 35 U.S.C. §112, second paragraph as being indefinite. Claims 1, 2, 6 – 14, 18 – 21 and 23 – 31 were rejected under 35 U.S.C. §102(b) as being anticipated by Ishihata et al. Claims 3 – 5, 15 – 17 and 22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Ishihata et al.

Examiner Zhe is greatly thanked for the telephone interview on March 27, 2008. In that interview, attorney Emile and Examiner Zhe discussed amended Claim 1 and Ishihata et al., the applied reference. Particularly, attorney Emile contrasted the addition of the designated process for ascertaining when all elements of the first array are updated and for issuing the release instruction to the applied reference of using registers and combinational logic to do so. Further, attorney Emile pointed to paragraphs [0072] and [0076] for support of the added limitations. No agreement was reached.

In response to the 101 rejection of Claims 14 – 21, Applicants have amended the pre-amble of independent Claim 14 to include "at least one processor for processing instructions to synchronize" and "the instructions comprising instructions to use:"

Support for the added limitations can be found in Fig. 1 where a plurality of processors 16 is displayed (see also paragraph [0051]), which any one then can be used to process code instructions in Figs. 3 – 6 as well as the different algorithms in paragraphs [0075] and [0081] – [0083]. Thus, no new matter is introduced in the Application by those added limitations.

By the added limitations, Applicants believe that the 101 rejection has been overcome and kindly request its withdrawal.

Regarding the 112 rejection of Claims 1, 14, 28 and 30 due to whether the process associated with an element in the first array is the same process associated with an element in the second array, Applicants have replaced the

limitations "at least two processes where each process is associated with an element of a first array and with an element of a second array" with a variable N, where N >= 2. Each i^{th} element of the first array is associated with an i^{th} process which is also associated with an i^{th} element of the second array, where 1 = i <= N. Thus, the same process is associated with a particular element of both the first and second arrays as i goes from 1 to N.

Applicants believe that by this amendment, the 112 rejection of independent Claims 1, 14, 28 and 30 has been overcome and kindly request its withdrawal.

In regard to the 112 rejection of Claims 24 and 25 due to the language in the last element of the claims, Applicants have deleted the offending language (i.e., ~~for said switch to said release state until detecting said release state~~). Thus, withdrawal of the rejection is kindly requested.

Regarding the 112 rejection of Claims 1, 14, 24 – 26, 28 and 30 due to the un-clarity of how the steps in the claims result in the synchronization of the concurrently running processes, Applicants have now used a first phase and a second phase and specify that the synchronization occurs at the first phase. Applicants have also specified that the elements of the second array, which are initialized to a hold state, are used to hold their associated processes that have completed the first phase at the first phase until they receive a release signal to release the processes to proceed to the second phase. Applicants have also specified that the release signal is issued when all concurrently running processes have completed the first phase.

Support for the added limitations can be found in paragraph [0055], among others, where it is disclosed that concurrently running processes are synchronized at barrier B1 of phase 1 before proceeding to phase 2. Further, in paragraph [0072], among others, it is disclosed that each concurrently running process will decrease an element of a first array from "1" to "0" and proceed to spin on a sensor until the sensor changes to a leave state. At that point the concurrently running processes will then proceed to the next phase.

Consequently, no new matter is added to the Application by the added limitations. Hence, Applicants kindly request withdrawal of the 112 rejection of Claims 1, 14, 24 – 26, 28 and 30.

Due to the amendment of the independent claims (i.e., Claims 1, 14, 24 – 26, 28 and 30), dependent Claims 2 – 8, 10, 15 – 19, 22, 23, 27, 29 and 30 are amended both for antecedent basis and to better claim the invention. Claims 9, 12, 13, 20 and 21 are canceled.

By this amendment, Claims 1 – 8, 10, 11, 14 – 19 and 22 - 31 remain pending in the Application. For the reasons stated more fully below, Applicants submit that the pending claims are allowable over the applied reference. Hence, reconsideration, allowance and passage to issue are respectfully requested.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

> 1. A method of synchronizing N concurrently running processes in a data processing system at a first phase before allowing the N processes to proceed to a second phase, N >= 2, comprising:
>
> providing a first array of N elements initialized each to a first state wherein each i^{th} element of said first array is associated with an i^{th} concurrently running process which will update the i^{th} element of the first array to a second state in response to completing the first phase, where 1 = i <= N;
>
> providing a second array of N elements initialized each to a hold state wherein each i^{th} element of said second array is associated with the i^{th} concurrently running process associated with the i^{th} element of the first array and is used to hold the i^{th} associated concurrently running process at the first phase, and is enabled to switch, in response to receiving a release instruction, to a release state to release the i^{th} associated concurrently running process to proceed to the second phase; and
>
> ***using a designated process configured to ascertain when the N elements of the first array are updated to the second state to issue the release instruction to allow the N processes to proceed to the second phase***. (Emphasis added.)

Applicants submit that the claims, as presently drafted, are patentable over the applied reference.

Ishihata et al. purport to teach a synchronization control system in a parallel computer. According to the teachings of Ishihata et al., the synchronization control system uses a synchronization request register, a synchronization detecting register, a status request register and a status detecting register for each processor of the parallel computer. The synchronization control system further uses two processing phases. The first processing phase is when the processors are actually processing data and the second processing stage is when the processors are processing a message. When a processor reaches a synchronization point, it requests synchronization by entering a logical "1" in its synchronization request register. When all the processors have done so, the "1s" in the synchronization request registers are ANDED and the result is used to reset the synchronization detecting register of each one of the processors.

When a processor finishes executing data, it will process the message. The message is sent from one processor to another. The message ensures that the system is not in error (i.e., no one processor is in error by being in a loop or driving a zero etc.). When a processor finishes handling a message and passes the message to the next processor, it sets its status request register by entering a logical "1" therein. As in the case of the synchronization request, when all the processors have set their status request register, the "1s" are ANDED and the result used to set the status detecting register of the processors.

When both the synchronization detecting requests and the status detecting registers are set, the system is synchronized and proceeds to the next processing stage.

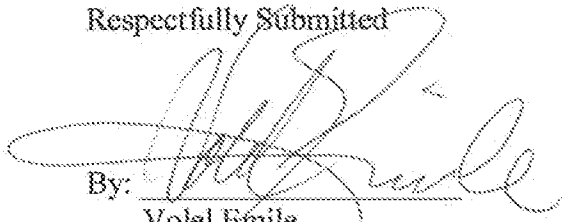
Thus, Ishihata et al. use combinational logic technology to determine when all the **processors** have completed a first phase and are ready to proceed to the next phase. By contrast the present invention uses a designated process

to determine when concurrently running **processes** have completed a first phase and are ready to proceed to the next phase.

Therefore, Ishihata et al. do not teach, show or suggest ***using a designated process configured to ascertain when all N elements of the first array are updated to the second state to issue the release instruction to allow the N processes to proceed to the second phase*** as claimed.

Hence, Applicants submit that Claim 1, as well as its dependent claims, is allowable over the applied reference. Independent Claims 14, 24 – 26, 28 and 30, along with their dependent claims, which incorporate the emboldened-italicized limitations of the above-reproduced Claim 1, in one fashion or another, are also allowable over the combined teachings of the applied references. Consequently, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully Submitted



By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969